HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN-FH
MÜNCHEN

Goldbach & Euler

Rolf Schirm,
Markus Mohr,
Franz Mathauser

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gelöste Probleme (19)

Mengen

- #11172 Relational Operator

Teilbarkeit

- #543 Goldbach's Conjecture
- #686 Goldbach‘s Conjecture II
- #382 Perfection
- #583 Prime Factors
- #10699 Problem D - Count the factors

Zahlen

- #10579 FibonacciNumbers
- #10101 Bangla Numbers
- #10622 Perfect P-th Powers

Zeichenketten

- #444 Encoder and Decoder
- #445 Marvelous Mazes
- #458 The Decoder
- #483 Word Scramble
- #494 Kindergarten Counting Game
- #10424 Love Calculator
- #11233 Deli Deli

Formeln

- #541 Error Correction
- #10370 Above Average
- #11219 How old are you?

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Schönste Probleme

#11172 Relational Operator

#10857 Easter-Eggs

#541 Error Correction

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Schönste Programme

#10699 Problem D - Count the factors (Mohr, Schirm, Mathauser)

Runtime: 0.060 s

#10970 Big Chocolate (Leib, Tschannerl)

Runtime: 1.048 s

#10493 Cats with or without hats (Leib, Tschannerl)

Runtime: 0.100 s

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Prime Factors (#583)

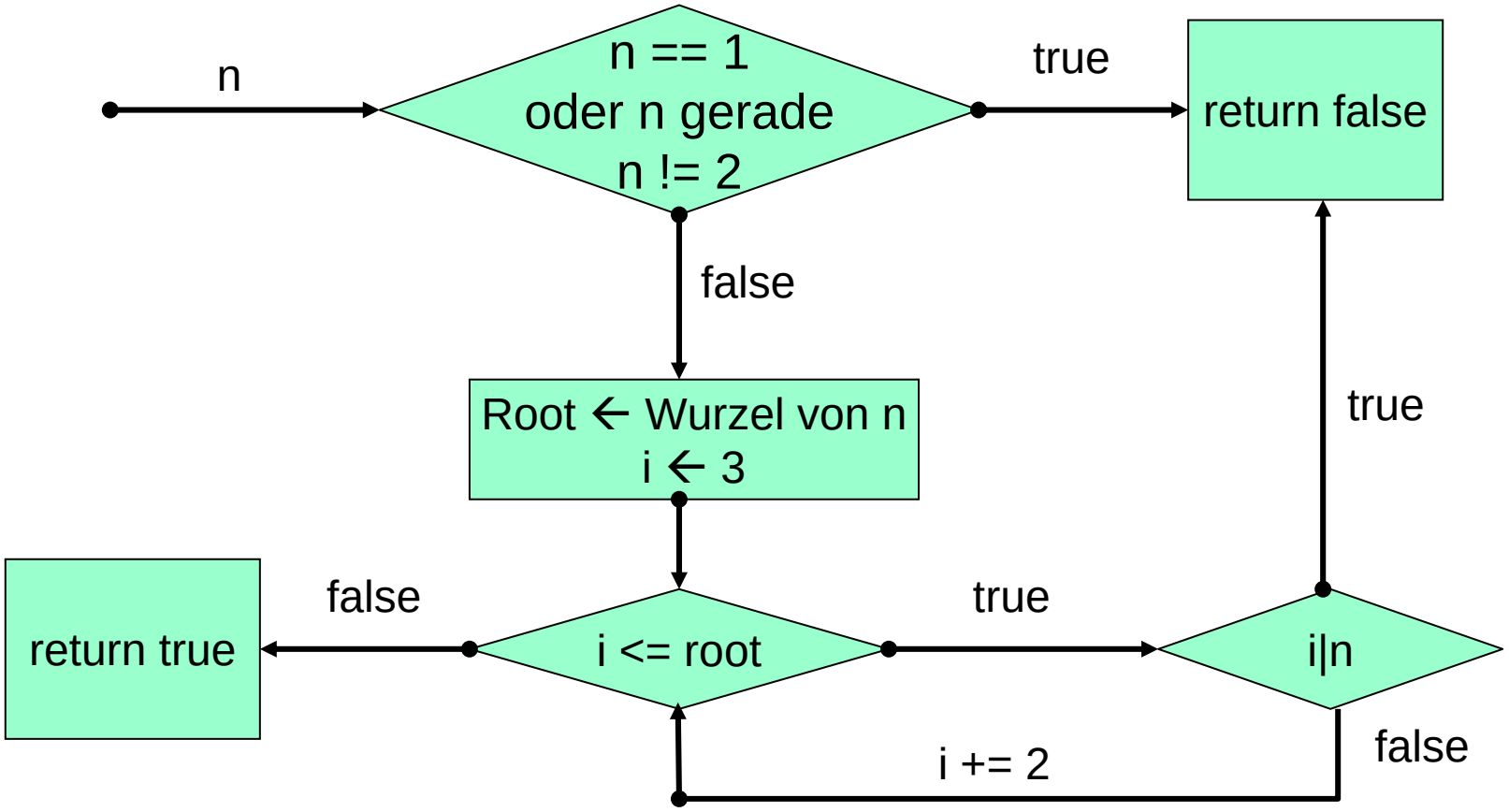
Sei *n* eine natürliche Zahl.

Eine Zahl *p* heißt *Primfaktor* von *n*,
wenn *p* ein Teiler von *n* ist und
p eine Primzahl ist.

Quelle: Wikipedia

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

isPrim Methode (Struktogramm)



HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

isPrim Methode (Java)

```
public static boolean isPrim(int n) {
        if(n == 1 || (n%2 == 0 && n != 2))
                return false;

        short root = (short)Math.sqrt(n);
        for (int i = 3; i <= root; i=i+2)
                if (n % i == 0)
                        return false;
        return true;
}
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Prime Factors (#583)

Input:

Jede Zeile enthält ein Integer g von der Größe $-2^{31} < g < 2^{31}$

Ohne –1 und 1

Input wird terminiert durch 0

Output:

Primfaktorzerlegung $g = f_1 \times f_2 \times \ldots \times f_n$

falls $g < 0$: $g = -1 \times f_1 \times f_2 \times \ldots \times f_n$

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Prime Factors (#583)

-190 = -1 x 2 x 5 x 19

-192 = -1 x 2 x 2 x 2 x 2 x 2 x 2 x 3

-193 = -1 x 193

-194 = -1 x 2 x 97

196 = 2 x 2 x 7 x 7

197 = 197

200 = 2 x 2 x 2 x 5 x 5

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Lösungsansatz (#583)

1. Array mit Primzahlen bis zur Wurzel von **MAX_INTEGER** füllen
2. Falls Zahl negativ → **Zahl = (-1)*Zahl** und (-1) ins Faktoren-Array hinzufügen
3. Schleife im Primzahlen-Array bis zur Wurzel der Zahl:

   falls **Zahl % Primzahl == 0**: Eintrag ins Faktoren-Array und **Zahl = Zahl/Primzahl**
4. Wenn **Zahl > 1** nach durchlauf der Schleife: Zahl ins Faktoren-Array eintragen
5. Ausgabe der Faktoren und prüfen ob nachfolgende Zahl gleich ist

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Prime Factors (#583)

```
for (int i = 0; i < anz && prime[i]*prime[i] <= buf; i++)
        while(buf % prime[i] == 0) {
            buf/=prime[i];
            factors[count++]=prime[i];
        }

if (buf != 1)
        factors[count++]=buf;
```

Prime Factors (#583)

Accepted!

RUNTIME: `2.530`

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Historisch (#543)

In 1742, Christian Goldbach, a German amateur mathematician, sent a letter to Leonhard Euler in which he made the following conjecture:

Every number greater than 2 can be written as the sum of three prime numbers.

Goldbach was considering 1 as a primer number, a convention that is no longer followed. Later on, Euler re-expressed the conjecture as:

Every even number greater than or equal to 4 can be expressed as the sum of two prime numbers.

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach‘s Conjecture (#543)

Input:

Jede Zeile enthält ein Integer n von der Größe $6 \leq n < 1000000$

Input wird terminiert durch 0

Output:

- n = a + b, wobei b – a maximal
- „Goldbach's conjecture is wrong."

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach‘s Conjecture (#543)

8 = 3 + 5

20 = 3 + 17 = 7 + 13

42 = 5 + 37 = 11 + 31 = 13 + 29 = 19 + 23

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Lösungsansatz 1 (#543)

1. Array mit Primzahlen bis zur größten eingegebenen Zahl
2. Schleife über die Primzahlen bis zur Zahl

   Test ob isPrim(n-Primzahl) („isPrim“=Methode)

   Falls true : Ausgabe und Abbruch der Schleife
3. Nach durchlauf der Schleife ohne bisherige Ausgabe:

   Ausgabe: „Goldbach's conjecture is wrong.“

Goldbach‘s Conjecture (#543)

Accepted!

RUNTIME: `2.320`

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Lösungsansatz 2 (#543)

1. Array mit n Primzahlen (bis zur max. möglichen Zahl)
2. Schleife über die Primzahlen bis Zahl/2

   Suche mit binarySearch im Prim Array

   Falls >= 0 : Ausgabe und Abbruch der Schleife
3. Nach durchlauf der Schleife ohne bisherige Ausgabe:

   Ausgabe: „Goldbach's conjecture is wrong.“

!! Verbesserte Performance !!

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Java-Code (#543)

```
for(int i = 0; (first = prime[i]) <= (second = zahl - first); i++) {
        if (Arrays.binarySearch(prime, second) >= 0) {
                primePair=true;
                break;
        }
}
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach‘s Conjecture 2 (#686)

Änderungen zu Goldbach‘s Conjecture:

Kleinerer Zahlenbereich: $n < 32768\ (2^{15})$

Kein Konkretes Primzahlpaar gesucht, sondern die

Anzahl aller möglichen Paare

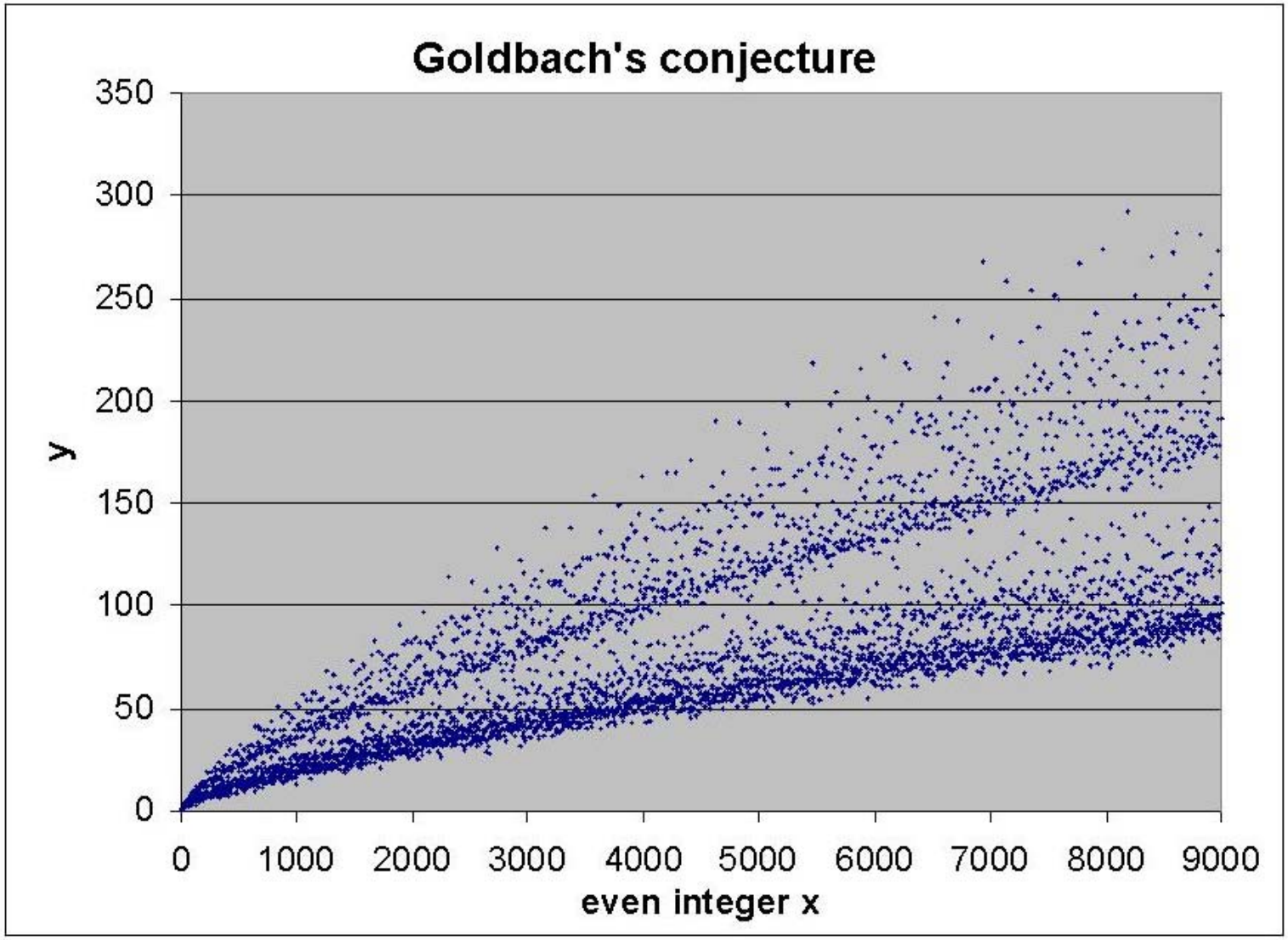
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach‘s Conjecture 2 (#686)

8 = 3+5 → 1

20 = 3+17 = 7+13 → 2

42 = 5+37 = 11+31 = 13+29 = 19+23 → 4



Quelle : Wikipedia

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Goldbach‘s Conjecture 2 (#686)

Änderungen zu Goldbach‘s Conjecture:

```
for(int i = 0; (first = prime[i]) <= (second = zahl - first); i++)
    if (Arrays.binarySearch(prime, second) >= 0)
        count++;

System.out.println(count);
```

Goldbach‘s Conjecture 2 (#686)

Accepted!

RUNTIME: `0.180`

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach and Euler (#10311)

Input:

- 100.000 Testfälle
- Jede Zeile enthält ein Integer n kleiner 100.000.000
- 40 MB Memory Limit
- 10 sec Time Limit

Output:

- „n is the sum of a and b“, wobei b – a minimal aber positiv
- „n is not the sum of two primes!"

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach and Euler (#10311)

Input:	**Output:**
11	11 is not the sum of two primes!
12	12 is the sum of 5 and 7.
13	13 is the sum of 2 and 11.
14	14 is the sum of 3 and 11.
15	15 is the sum of 2 and 13.
16	16 is the sum of 3 and 13.
42	42 is the sum of 19 and 23.

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach and Euler (#10311)

set[0] = *-1601558355*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	1	**1**	0	0

set[1] = *673221152*

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

```
public void set(int pos, boolean value)
{
        if(value)
            set[pos>>5] |= (1<<(pos & 0x0000001F));
         else
            set[pos>>5] &= ~(1<<(pos & 0x0000001F));
}                                                    //1F = 31 (dez.)
```

pos>>5 = Rechts-Shift um fünf stellen → pos / 32

= Wert des Array Index für gesuchte pos

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

Beispiel: mit pos 50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0

& - Binär Verknüpfung

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

=

18 (dez)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

Sieb des Eratosthenes:

Alle Zahlen von 2 bis zur Wurzel der maximalen Zahl werden durchlaufen:

- Unmakiert: Primzahl, alle Vielfachen makieren
- Makiert: keine Primzahl

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach and Euler (#10311)

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Goldbach and Euler (#10311)

```
for(int i=2; i <= squareRoot; i++)
        if(isPrime.get(i))
                for(int j=i+i;  j <= MAX_PRIME; j+=i)
                        isPrime.set(j,false);
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

Im Hauptprogramm

- Zahl einlesen
- Ungerade → erster Summand muss 2 sein
- Gerade → Suche des Primzahlenpaars

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

```
if(n % 2 == 1)
        if(n != 1 && isPrime.get(n-2))
                firstPrime = 2;
else
    for(int i = n/2; i >= 2; i--)
        if(isPrime.get(i) && (n-i != i) && isPrime.get(n-i))
        {
                firstPrime = i;
                break;
        }
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Goldbach and Euler (#10311)

Ausgabe

- erster Summand gleich Start-Wert (-1):

„n is not the sum of two primes!„

- sonst: Ausgabe der beiden Summanden

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Gliederung

- Prime Factors (#583)
- Goldbach‘s Conjecture (#543)
- Goldbach‘s Conjecture II (#686)
- Goldbach and Euler (#10311)
- How old are you? (#11219)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

How old are you? (#11219)

Input:

- Anzahl der Testfälle
- aktuelles und Geburtsdatum im Format:
  DD/MM/YYYY

Output: #Testcase

- Alter < 0: „Invalid birth date“
- Alter > 130: „Check birth date“
- Alter

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

How old are you? (#11219)

4

01/01/2007
10/02/2007

09/06/2007
28/02/1871

12/11/2007
01/01/1984

28/02/2005
29/02/2004

Case #1: Invalid birth date

Case #2: Check birth date

Case #3: 23

Case #4: 0

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN-FH MÜNCHEN

Lösungsansatz (#11219)

- Eingelesene Strings in int[] umformen
- Berechnung der Tage zwischen den Daten

  static long daysBetween(int[] start, int[] end)
- Berechnung der Jahre aus den oberen Ergebnis

  static int calcYears(long between, int[] birthdate)
- Ausgabe

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

How old are you? (#11219)

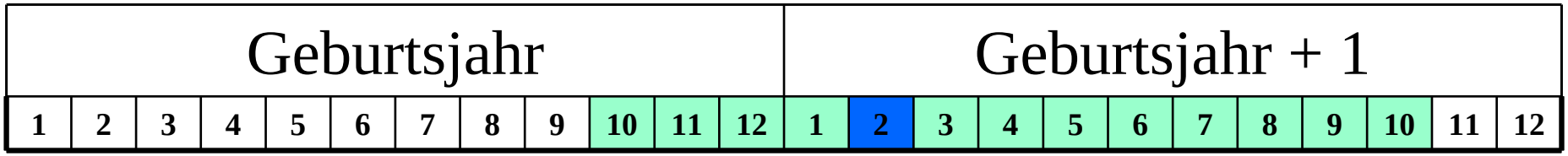
```
public static long daysBetween(int[] start, int[] end) {
        long startDays = start[0];
        long endDays = end[0];

        if (start[2] > end[2])
        for(int i = end[2] ; i < start[2]; i++)
            startDays += (Schaltjahr(i)) ? 366 : 365;

        if (start[2] < end[2])
        for(int i = start[2] ; i < end[2]; i++)
            endDays += (Schaltjahr(i)) ? 366 : 365;
```

```
if (start[1] > end[1])
        for(int i = end[1] ; i < start[1]; i++)
            if(i == 2)
                startDays += (Schaltjahr(start[2])) ? 29 : 28;
            else
                if(i == 1 || i == 3 || i == 5 || i == 7 || i == 8 || i == 10)
                    startDays += 31;
                else
                    startDays += 30;
        if (start[1] < end[1])
        for(int i =start[1] ; i < end[1]; i++)
            if(i == 2)
                endDays += (Schaltjahr(end[2])) ? 29 : 28;
            else
                if(i == 1 || i == 3 || i == 5 || i == 7 || i == 8 || i == 10)
                    endDays += 31;
                else
                    endDays += 30;
        return endDays - startDays;
    }
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

How old are you? (#11219)

```
public static int calcYears(long between, int[] birthDate)   {
    int age = 0;
    if (birthDate[1] > 2)
            birthDate[2] += 1;
```




```
                between -= 366;
            else
                between -=365;
            age++;
            birthDate[2]++;
            SchaltJahr = Schaltjahr(birthDate[2]);
    }
    return age;
}
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

How old are you? (#11219)

```
public static int calcYears(long between, int[] birthDate)    {
    int age = 0;
    if (birthDate[1] > 2)
        birthDate[2] += 1;
    boolean SchaltJahr = Schaltjahr(birthDate[2]);
    while(SchaltJahr && between >= 366 || !SchaltJahr && between >= 365) {
        if (SchaltJahr)
            between -= 366;
        else
            between -=365;
        age++;
        birthDate[2]++;
        SchaltJahr = Schaltjahr(birthDate[2]);
    }
    return age;
}
```

How old are you? (#11219)

Accepted!

RUNTIME: `0.170`

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN - FH MÜNCHEN

Vielen Dank für Ihre Aufmerksamkeit